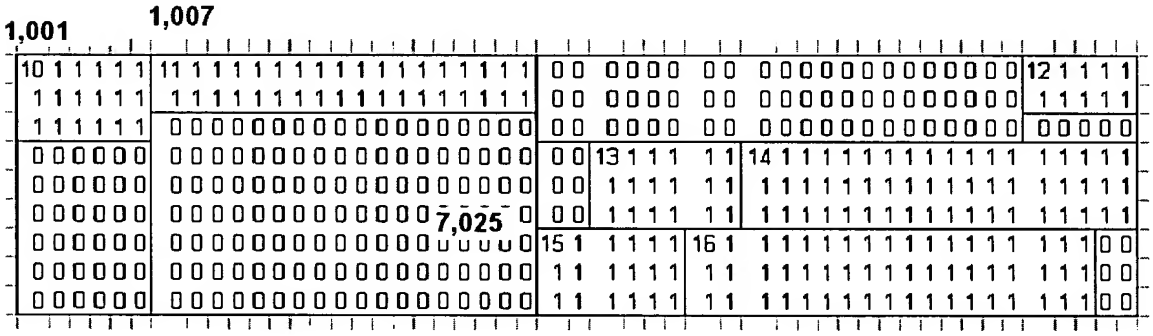
Appendix A

Given the position and size of control objects in the grid, the following algorithm constructs a HTML table to display the controls. A HTML table is preferable since not all browser support pixel-precise style sheet correctly. The position of a control is defined by the (row, column) pair, where the row and column correspond to the position of the control within the grid displayed below. ROWSPAN identifies how many grid columns tall the control should be. COLSPAN identifies how many grid columns wide the control should be.

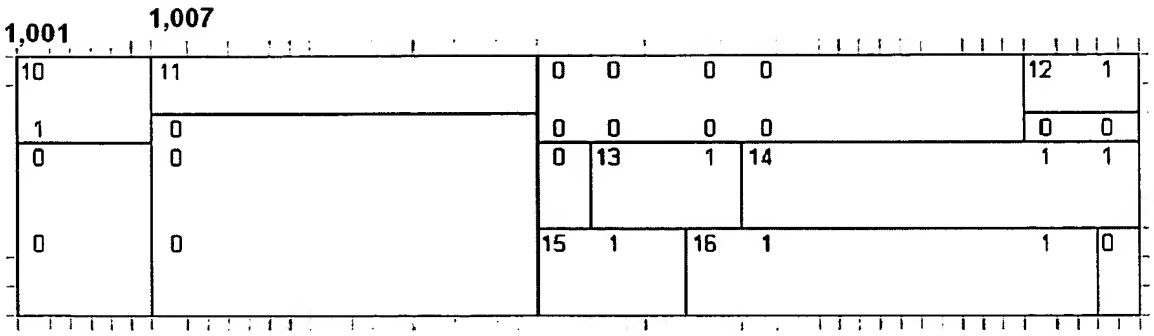
The following is exemplary pseudocode for creating a HTML table:

*1st pass:
Loop through all control objects and mark the grid with 0s, 1s and control identifiers. During this loop the maximum height and width of the grid will be determined.

The 1's on the grid correspond to space taken up by the span of the controls.
The 0's on the grid correspond to empty space that needs to be filled for spacing.
The pseudo control identifier = control identifier + 10, is placed at the left –upper corner of the grids where the control resides.
The lines correspond to how the grid will be broken up into pieces to be displayed as an HTML table.



*2nd pass:
Create two integer arrays, one to represent the number of rows and another for the columns. Fill the integer array with 1's at every place where a control starts or stops, to determine every row and column that is a starting point. All the spaces that are left with a 0 can be removed as unnecessary rows and columns. For each cell that is removed, its starting position cell will be incremented by 1, this will allow us to preserve the height and width of the grid using Cascading Style Sheet (CSS) later. After the number of rows and columns that can be condensed is determined, it is necessary to update the ROWSPAN, COLSPAN, and (row, column) pair to reflect the condensed grid size. The new grid would look like this:



Then define a matrix *M* of 0s with the height and width matching the max height and width determined from the 1st pass. Loop through the controls. Determine the Row/Column on the matrix to set as (control identifier + 10), use the corresponding ROWSPAN and COLSPAN to determine the 1's in the array to set.

3rd pass:
Setup 1st HTML row for table cells of fixed 8 pixels in width.
Loop through entire matrix *M* checking values:
Case 0 :

Scan the matrix right until the end of the column is reached or a non 0 is encountered. Go down one matrix position if it is not the end and if the number is a 0, then repeat right scan, and if the calculated COLSPAN from this is lower than the previous one, use the lower one. Repeat this scenario until it is not possible to go any lower. We now know how large of a TD cell to create for the HTML node. Set all the matrix positions used to 1.

Case 1 :

Do Nothing.

Default :

Create control node with correct spans on surrounding html. The control identifier will be located at the value of the current matrix item minus 10.